IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF – 37 C.F.R § 1.192

U.S. Patent Application 10/709,415 entitled:

"Self-Adaptive Prefix Encoding for Stable Node Identifiers"

Real Party in Interest: International Business Machines Corporation

Related Appeals and Interferences:

None

Status of Claims:

Claims 1-16 are pending.

Claims 1-16 are rejected.

Claims 1-4, 9 and 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,889,226 (O'Neil et al.), in view of US Patent 6,263,332 (Nasr et al.), and further in view of US Publication 2002/0120690 (Hayton).

Claims 5-8 and 10-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Publication 2003/0110150 (O'Neil et al.), in view of US Publication 2002/0120679 (Hayton et al.).

Claims 1-16 are hereby appealed.

Status of Amendments:

The amendment after final filed 7/18/2007 has been entered for the purposes of appeal.

Summary of Claimed Subject Matter:

(NOTE: All citations are made from the associated Pre-Grant Publication 2006/0004858)

According to independent **claim 1**, the present invention provides for a computer-based method comprising the steps of: (a) choosing an initial base length with which to encode local identifiers ***(see paragraphs [0034], [0035], [0038], and [0039])***, (b) assigning a value of zero as

a node identifier to a root node in a logical tree ***(see paragraph [0032] and figures 1-5)***, (c) sequentially assigning to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned in increasing value from leftmost children to rightmost children ***(see paragraph [0032] and figures 1-5)***, (d) assigning node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned ***(see paragraphs [0032], [0033] and figures 1-5)***, and (e) extending said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers ***(see paragraphs [0034], [0035], [0036], [0037], [0038] and [0039] and figures 1-5)***.

According to independent **claim 9**, the present invention provides for an article of manufacture comprising a computer readable storage medium having computer readable program code embodied therein comprising modules implementing code to: (a) choose an initial base length with which to encode local identifiers ***(see paragraphs [0034], [0035], [0038], [0039] and [0049])***, (b) assign a value of zero as a node identifier to a root node in a logical tree ***(see paragraphs [0032] and [0049] and figures 1-5)***, (c) sequentially assign to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned in increasing value from leftmost children to rightmost children ***(see paragraphs [0032] and [0049] and figures 1-5)***, (d) assign node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned ***(see paragraphs [0032], [0033], and [0049] and figures 1-5)***, and (e) extend said initial base length when local identifier encoding

combinations are exhausted before all descendants are assigned local identifiers ***(see paragraphs [0034], [0035], [0036], [0037], [0038] and [0039] and figures 1-5)***.

According to independent **claim 16**, the present invention provides for a computer-based method comprising the steps of: (a) choosing an initial base length with which to encode local identifiers ***(see paragraphs [0034], [0035], [0038], [0039] and [0049])***, (b) assigning a value of zero as a node identifier to a root node in a logical tree ***(see paragraphs [0032] and [0049] and figures 1-5)***, (c) sequentially assigning to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned said even values based on variable-length binary string encoding and said local identifiers are assigned in increasing value from leftmost children to rightmost children ***(see paragraphs [0032], [0039], and [0049] and figures 1-5)***, (d) assigning node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned ***(see paragraphs [0032], [0033], and [0049] and figures 1-5)***, and (e) extending said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers ***(see paragraphs [0034], [0035], [0036], [0037], [0038] and [0039] and figures 1-5)***.

Grounds of Rejection to be Reviewed on Appeal:

Claims 1-4, 9 and 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,889,226 (O'Neil et al.), in view of US Patent 6,263,332 (Nasr et al.), and further in view of US Publication 2002/0120690 (Hayton). Claims 5-8 and 10-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Publication 2003/0110150 (O'Neil et al.), in view of US Publication 2002/0120679 (Hayton et al.).

Claims 1-16 are hereby appealed.

Was a proper rejection made under 35 U.S. C. §103(a) using existing USPTO guidelines?

ARGUMENT:

Was a proper rejection made under 35 U.S. C. §103(a) using existing USPTO guidelines?

Claims 1-4 and 9 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,889,226 issued to O'Neil et al. (hereafter O'Neil), in view of U.S. Patent No. 6,263,332 issued to Nasr et al. (hereafter Nasr), and further in view of U.S. Publication No. 2002/0120679 issued to Hayton et al. (hereafter Hayton).

To establish a prima facie case of obviousness under U.S.C. §103, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. Additionally, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure (In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

O'Neil provides for a technique to represent hierarchical data in a non-hierarchical data structure, wherein the hierarchical data (e.g., XML data) can be viewed as having a "tree" structure, and each node in the tree is assigned a position identifier that represents both the depth

level of the node within the hierarchy, and its ancestor/descendant relationship to other nodes. According t O'Neil, the data represented by each node, as well as its position identifier, is stored in a row of a relational database, thereby capturing the hierarchical structure of the data in such relational database. O'Neil also provides for compressed storage of position identifiers in a format that allows an efficient bytewise comparison of position identifiers to determine relative order and ancestry.

Nasr et al. teaches a computer-implemented method of retrieving information in a first markup language through a query engine and presenting the information in any required markup language is shown.

Hayton teaches the features of associating an element of a user-interface to a current state of a property of an application where the application has a plurality of components having properties and each property being identified with an identifier, and concatenating of a plurality of identifiers to identify the final state value of a parameter (e.g., salary of an employee).

Applicants' claim 1, by stark contrast, provides for a computer-based method comprising the steps of: (a) choosing an initial base length with which to encode local identifiers, (b) assigning a value of zero as a node identifier to a root node in a logical tree, (c) sequentially assigning to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned in increasing value from leftmost children to rightmost children, (d) assigning node identifiers by

concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned, and (e) extending said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers.

Applicants' claim 9 provides for an article of manufacture implementing the above-described method.

Applicants' claim 16, also by stark contrast, provides for a computer-based method comprising the steps of: (a) choosing an initial base length with which to encode local identifiers, (b) assigning a value of zero as a node identifier to a root node in a logical tree, (c) sequentially assigning to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned said even values based on variable-length binary string encoding and said local identifiers are assigned in increasing value from leftmost children to rightmost children, (d) assigning node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned, and (e) extending said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers.

With respect to independent claims 1, 9, and 16, the Examiner states on pages 5-6 and 8-9 of the Final Office Action of 04/18/2007 that:

> "O'Neil does **not** disclose:
>
> a. assigning a value of zero as a node identifier to a root node in a logical tree
>
> b. assigning node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned". (emphasis added).

Further, with respect to independent claims 1, 9, and 16, the Examiner further states on the same pages of the Final Office Action of 04/18/2007 that:

> "The combination of Nasr and O'Neil does **not** disclose:
>
> d. assigning to all subsequent nodes, node identifiers generated by a concatenation of local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned". (emphasis added).

Applicants agree with both of the above conclusions by the Examiner that the neither the O'Neil reference nor the Nasr reference teaches claim 1 and 9's feature of **assigning node**

identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned.

However, Applicants respectfully disagree with the Examiner conclusion that such a feature of claim 1, 9, and 16 is remedied by the Hayton reference.

Specifically, the Examiner cites lines 3-9 in paragraph **[0052]** of Hayton as teaching such a feature. The Examiner's citation of paragraph **[0052]** is reproduced below:

> "[0052] ... The property path defines a path through the application 26 from a root component 38 to the particular application component 34, and a property 38 itself of that application component. ..."

The above citation in Hayton is in reference to Hayton's Figure 1, which is reproduced below:

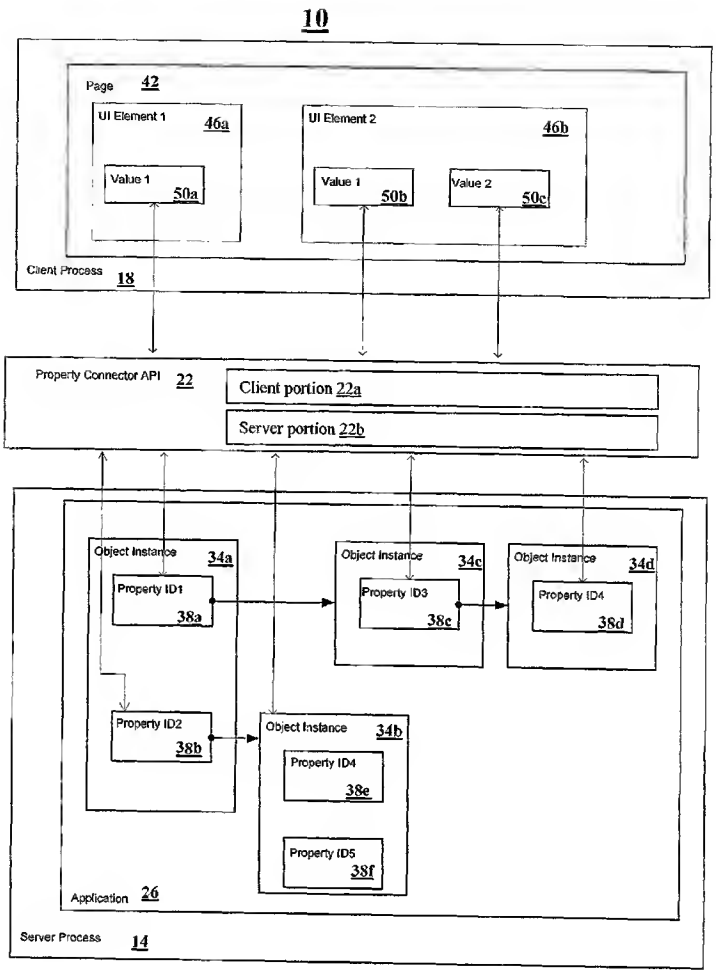

FIG. 1

By Hayton's own admission, FIG. 1 merely represents a block diagram of a system 10 for "**communicating changes between a user-interface and an executing application, using property paths**". Further, the concatenation of such property paths merely leads to the identification of "**a current state of the property at the end of the path**".

The board is respectfully directed to review paragraph [**0053**] of the Hayton reference which outlines, with specificity, the reasoning behind such concatenation. Specifically, in paragraph [**0053**] of Hayton, it specifically states that "**The property connector API 22**

determines the present state points to the application component 34d." It can be seen from Hayton's Fig. 1 reproduced above that such concatenation of paths is needed as the "Property ID1" points to "Property ID3", which in turn points to "Property ID4", wherein "Property ID4" belongs to object instance 34d. **It is only with such mapping and concatenation, the *final state value* is determined in Hayton's system/method**.

Hence, Applicants maintain that Hayton merely teaches **associating an element of a user-interface to a current state of a property of an application** where the application has a plurality of components having properties and each property being identified with an identifier (e.g., Property ID1, Property ID2, etc.) and **concatenating plurality of such identifiers to determine a final state value** (e.g., an actual value such as the salary of an employee).

Claims 1, 9, and 16, by contrast, teach the ***assignment of node identifiers* by concatenating local identifiers of all nodes along a path from a root node to a node *to which a node identifier is currently being assigned***. The Hayton reference fails to explicitly or implicitly mention ANY such ***assignment of a node identifier*** based on a **concatenation of local identifiers of all nodes along a path from the root node to a node *to which a node identifier is currently being assigned***.

For clarification of claims 1, 9, and 16, the Examiner is respectfully requested to review figure 4 of the application-as-filed, where Applicants' concatenation feature is outlined.

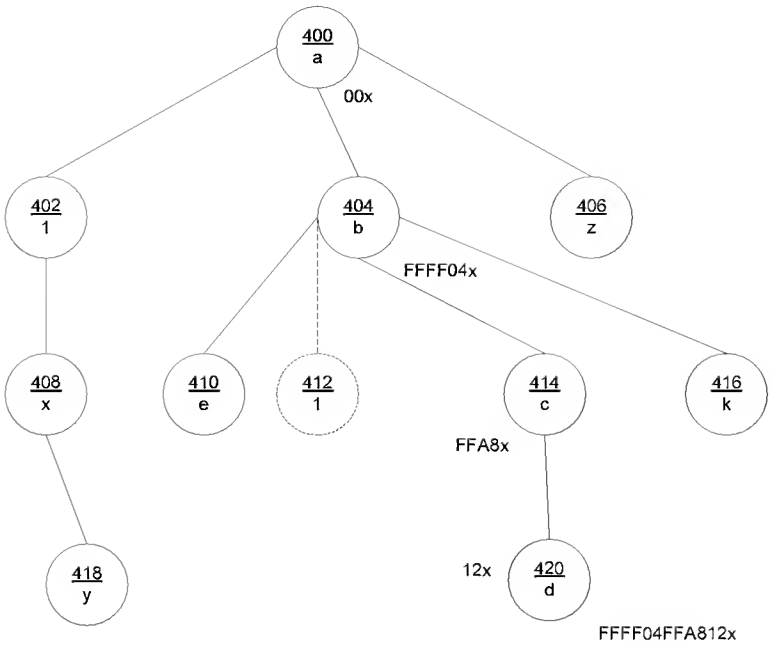

FIGURE 4

It can be seen from the above-figure that **what is concatenated as per Applicants' invention is the encodings for nodes *a*, *b*, *c*, and *d*, whose values are 0, FFFF04x, and FFA8x, and 12x**, respectively. **Such a concatenation results in a node id for *d* with a value of FFFF04xFFA8x12x**, which is a concatenation of the individual strings 0, FFFF04x, and FFA8x, and 12x. It should be emphasized that **such concatenation is specifically used in the assignment of node identifiers and CANNOT be equated to a series of pointers that are traversed to identify a *final state value***.

Hence, Applicants respectfully submit that Hayton's **concatenation dealing with identifying the final state value** CANNOT be equated to claim 1, 9, and 16's feature of **assigning a node identifier based on concatenation of local identifiers**.

Applicants also respectfully contend that the Hayton reference could not have been combined with O'Neil and Nasr by one of ordinary skill in the art. Specifically, although O'Neil and Nasr deal with structured documents, Hayton merely deals with "**APIs**" (Application Programming Interfaces) and "**user interfaces**". In Hayton's own words, their invention relates to "client-server networks" and refers to "client nodes" and "server nodes" (for example, see paragraphs [0002] through [0008]). The solitary mention of XML documents occur on in paragraph [0047], where Hayton mentions that that a component 34 could collectively represent nodes in an XML file and **makes no mention of prefix encoding node identifiers**. Hence, Applicants respectfully assert that one of ordinary skill in the art would have not been able to apply such teachings in prefix encoding methods. Further, even if it one of skill in the art were to combine the teachings of O'Neil, Nasr and Hayton, it is still respectfully maintained that the Hayton reference's concatenation deals with identifying a **final state value** and makes **no explicit or implicit mention of using such concatenation for prefix encoding, let alone assigning node identifiers based on concatenation of local identifiers**.

Applicants, therefore, submits that a prima facie case of obviousness has not been successfully established with respect to independent claims 1, 9, and 16. Applicants believe that independent claims 1, 9, and 16 are allowable over the cited art at least for the reasons set forth above. Hence, Applicants respectfully assert that an improper 35 U.S.C. §103 rejection was issued with regards to independent claims 1, 9, and 16.

The arguments presented above substantially apply to dependent claims 2-8, 10-15 as they inherit all the features of the claim from which they depend. Hence, Applicants believe that dependent claims 2-8 and 10-15 are allowable over the cited art at least for the reasons set forth above. Hence, Applicants respectfully assert that an improper 35 U.S.C. §103 rejection was issued with regards to dependent claims 2-8 and 10-15.

SUMMARY

As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of applicant's presently claimed invention, nor render them obvious. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

As this Appeal Brief has been timely filed within the set period of response, no fee for extension of time is required. However, the Commissioner is hereby authorized to charge any deficiencies in the fees provided, including extension of time, to Deposit Account No. 09-0460.

Respectfully submitted by
Applicant's Representative,

/ramraj soundararajan/

Ramraj Soundararajan
Reg. No. 53,832

IP Authority, LLC.
9435 Lorton Market Street
#801
Lorton, VA 22079
(571)642-0033

September 18, 2007

Claims Appendix:

1. (Currently Amended) A computer-based method comprising the steps of:

a. choosing an initial base length with which to encode local identifiers,

b. assigning a value of zero as a node identifier to a root node in a logical tree,

c. sequentially assigning to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned in increasing value from leftmost children to rightmost children,

d. assigning node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned, and

e. extending said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers.

2. (Currently Amended) The computer-based method of claim 1, wherein inserting a node into an existing tree does not require change to existing node identifiers.

3. (Currently Amended) The computer-based method of claim 1, wherein a node is inserted between a first node and a second node having consecutive local identifiers.

4. (Currently Amended) The computer-based method of claim 3, wherein said inserted node is assigned a local identifier having a string length longer than string length of said first node.

5. (Currently Amended) The computer-based method of claim 1, wherein assigning said node identifier to an inserted node comprises the following steps:

a. determining whether node to be inserted is inserted as a first child, between two existing siblings, or as a last child under a single parent node,

b. when said node to be inserted is inserted as a first child under said single parent node,

i. checking last byte of an existing first child,

ii. when the value of said last byte is not the smallest even number, then an even number greater than zero and less than the value of said last byte is selected to generate a local identifier of said node to be inserted, else

iii. when the value of said last byte of an existing first child is the smallest even number, generating a local identifier for said node to be inserted by replacing said last byte of said existing first child by an odd number to generate a local identifier and extending node identifier of said existing first child by a byte having a value of any arbitrary even number,

c. when said node to be inserted is inserted between two existing siblings under said single parent node, determining whether the string length of node identifier of said first sibling is less than, equal to, or greater than the string length of node identifier of said second sibling, else

d. when said node to be inserted is inserted as a last child after all other children under said single parent node, assigning to said node to be inserted an even local identifier greater than that of existing last child under said single parent node, and

generating a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

6. (Currently Amended) The computer-based method of per claim 5, when said node to be inserted is inserted between two existing siblings under said single parent node and when the string length of local identifier of said first sibling is less than the string length of the local identifier of said second sibling,

a. checking when local identifier of said first sibling is the last available encoding value having a string length of the local identifier of said first sibling and being smaller in value than said local identifier of said second sibling,

b. when said local identifier of said first sibling is the last combination having a string length of the local identifier of said first sibling that is smaller in value than said local identifier of said second sibling,

i. when the local identifier of said second sibling is not the first available identifier having the string length of the local identifier of said second sibling that is greater than the value of said local identifier of said first sibling; an even-valued local identifier being less in value than said local identifier of said second sibling and having string length of local identifier of said second sibling is generated and assigned, else

ii. generating a local identifier for said node to be inserted by replacing said last byte of said existing first child by an odd number and extending local identifier of said existing first child by a byte having a value of any arbitrary even number less in value than said last byte of said existing first child, and

generating a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

7. (Currently Amended) The computer-based method of claim 5, when said node to be inserted is inserted between two existing siblings under said single parent node and when the string length of the local identifier of said first sibling is equal to the string length of the local identifier of said second sibling,

a. when the value of the local identifier of said first sibling plus two is less than the value of the local identifier of said second sibling, a local identifier for said node to be inserted takes on an even value greater than or equal to the value of said local identifier of first sibling plus two and less than the value of the local identifier of said second sibling,

b. when the string length of the local identifier of said first sibling plus two is equal to the string length of the local identifier of said second sibling, then the string length of the local identifier for said node to be inserted is extended wherein the length of the local identifier for the newly inserted node is the string length of said second sibling plus one, and the value of the first string length of said first sibling bytes is the node identifier of said first sibling plus one, and the new byte is an arbitrary even number less than the value of said last byte of the node identifier of said second sibling, and

generating a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

8. (Currently Amended) The computer-based method of claim 5, when said node to be inserted is inserted between two existing siblings under said single parent node and when the string length of the local identifier of said first sibling is greater than the string length of the local identifier of said second sibling

a. when the local identifier of said second sibling is not the smallest value having the string length of said second sibling that is greater in value than the local identifier of said first sibling, then a local identifier having a string length of said second sibling and having even value smaller than the value of the last byte of the node identifier of said second sibling is generated and assigned else,

b. when the local identifier of said first sibling is not the largest value with the string length of the local identifier of said first sibling, one of the larger values for the new encoding is generated and assigned, else

c. extending the local identifier of said first sibling by a length, by setting the last byte to the highest odd number and the new byte to an even number less than the value of the last byte, and

generating a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

9. (Currently Amended) An article of manufacture comprising a computer readable storage medium having computer readable program code embodied therein comprising modules implementing code to:

a. choose an initial base length with which to encode local identifiers,

b. assign a value of zero as a node identifier to a root node in a logical tree,

c. sequentially assign to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned in increasing value from leftmost children to rightmost children,

d. assign node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned, and

e. extend said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers.

10. (Currently Amended) The article of manufacture of claim 9, wherein assigning a prefix encoded node identifier to an inserted node comprises modules implementing code to:

a. determine whether node to be inserted is inserted as a first child, between two existing siblings, or as a last child under a single parent node,

b. when said node to be inserted is inserted as a first child under said single parent node,

i. check last byte of an existing first child,

ii. when the value of said last byte is not the smallest even number, then an even number greater than zero and less than the value of said last byte is selected to generate a local identifier of said node to be inserted, else

iii. when the value of said last byte of an existing first child is the smallest even number, generate a local identifier for said node to be inserted by replacing said last byte of said existing first child by an odd number to

generate a local identifier and extending node identifier of said existing first child by a byte having a value of any arbitrary even number,

c. when said node to be inserted is inserted between two existing siblings under said single parent node, determine whether the string length of node identifier of said first sibling is less than, equal to, or greater than the string length of node identifier of said second sibling, else

d. when said node to be inserted is inserted as a last child after all other children under said single parent node, assign to said node to be inserted an even local identifier greater than that of existing last child under said single parent node, and

generate a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

11. (Currently Amended) The article of manufacture of claim 10, wherein when said node to be inserted is inserted between two existing siblings under said single parent node and when the string length of local identifier of said first sibling is less than the string length of the local identifier of said second sibling, comprises modules implementing code to:

a. check when local identifier of said first sibling is the last available encoding value having a string length of the local identifier of said first sibling and being smaller in value than said local identifier of said second sibling,

b. when said local identifier of said first sibling is the last combination having a string length of the local identifier of said first sibling that is smaller in value than said local identifier of said second sibling,

i. when the local identifier of said second sibling is not the first available identifier having the string length of the local identifier of said second sibling that is greater than the value of said local identifier of said first sibling; an even-valued local identifier being less in value than said local identifier of said second sibling and having string length of local identifier of said second sibling is generated and assigned, else

ii. generate a local identifier for said node to be inserted by replacing said last byte of said existing first child by an odd number and extending local identifier of said existing first child by a byte having a value of any arbitrary even number less in value than said last byte of said existing first child, and

generate a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

12. (Currently Amended) The article of manufacture of claim 10, wherein when said node to be inserted is inserted between two existing siblings under said single parent node and when the string length of the local identifier of said first sibling is equal to the string length of the local identifier of said second sibling, comprises modules implementing code to:

a. when the value of the local identifier of said first sibling plus two is less than the value of the local identifier of said second sibling, a local identifier for said node to be inserted takes on an even value greater than or equal to the value of said local identifier of first sibling plus two and less than the value of the local identifier of said second sibling,

b. when the string length of the local identifier of said first sibling plus two is equal to the string length of the local identifier of said second sibling, then the string length of the local identifier for said node to be inserted is extended wherein the length of the local identifier for the newly inserted node is the string length of said second sibling plus one, and the value of the first string length of said first sibling bytes is the node identifier of said first sibling plus one, and the new byte is an arbitrary even number less than the value of said last byte of the node identifier of said second sibling, and

generate a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

13. (Currently Amended) The article of manufacture of claim 10, wherein when said node to be inserted is inserted between two existing siblings under said single parent node and when the string length of the local identifier of said first sibling is greater than the string length of the local identifier of said second sibling, comprises modules implementing code to:

a. when the local identifier of said second sibling is not the smallest value having the string length of said second sibling that is greater in value than the local identifier of said first sibling, then a local identifier having a string length of said second sibling and having even value smaller than the value of the last byte of the node identifier of said second sibling is generated and assigned else,

b. when the local identifier of said first sibling is not the largest value with the string length of the local identifier of said first sibling, one of the larger values for the new encoding is generated and assigned, else

c. extending the local identifier of said first sibling by a length, by setting the last byte to the highest odd number and the new byte to an even number less than the value of the last byte, and

generate a node identifier by a concatenation of local identifiers of all nodes along a path from a root node to said node to be inserted.

14. (Currently Amended) The computer-based method of claim 1, wherein said assigned local identifiers are assigned values based on variable-length binary string encoding.

15. (Currently Amended) The article of manufacture of claim 9, wherein said assigned local identifiers are assigned values based on variable-length binary string encoding.

16. (Currently Amended) A computer-based method comprising the steps of:

a. choosing an initial base length with which to encode local identifiers,

b. assigning a value of zero as a node identifier to a root node in a logical tree,

c. sequentially assigning to descendants of a root node a local identifier having an even value and a length equal to said base length chosen in said choosing step, wherein said local identifiers are assigned said even values based on variable-length binary string encoding and said local identifiers are assigned in increasing value from leftmost children to rightmost children,

d. assigning node identifiers by concatenating local identifiers of all nodes along a path from a root node to a node to which a node identifier is currently being assigned, and

extending said initial base length when local identifier encoding combinations are exhausted before all descendants are assigned local identifiers.

Evidence Appendix

None

Related Proceedings Appendix

None